**PHP – Jedi – Danki Code**

**Aula03 - Mostrar na tela**

<?php

echo $\_SERVER['DOCUMENT\_ROOT'];

?>

<?php

echo '<pre>';

print\_r($\_SERVER);

echo '</pre>';

?>

**Aula04 - Variaveis**

<?php

$nome = 'Guilherme';

echo 'Meu nome é <br />';

echo $nome;

$nome = 'João';

echo '<br />';

echo 'Meu nome é <br />';

echo $nome;

?>

<?php

echo '<br />';

/\*String\*/

$nome = 'Guilherme';

/\*Valores inteiros\*/

$idade = 23;

echo $idade;

?>

<?php

/\*

Comentários para auxiliar

\*/

/\*String\*/

$nome = 'Guilherme';

/\*Número ou inteiro\*/

$idade = 23;

/\*Boolean\*/

$bool = true;

/\*Doubles\*/

$double = 10.09;

/\*Agora vou printar na tela minhas 2 variáveis\*/

/\*

echo 'Meu nome é'.$nome;

echo "Minha idade é ".$idade;;

\*/

echo $bool;

?>

**Aula05 - Constantes**

<?php

define('NOME', 'GUILHERME');

echo NOME;

echo '<br />';

define('DOMINIO', 'http://meudominio.com');

echo DOMINIO

?>

**Aula06 - Arrays**

<?php

/\*

Arrays = Variáveis com diversas chaves!

\*/

//$nome = array('Guilherme', 'João', 'Felipe');

//$nome[] = 'João';

//$nome[] = 'Felipe';

//$nome[0] = 'João';

//$nome[100] = 'Felipe';

//$nome1 = 'Guilherme';

//$nome2 = 'Joao';

//$nome3 = 'Felipe';

//$variaveis = ['Guilherme', 'João', 'Felipe'];

//$variaveis = array('Guilherme',23,true,10.09);

//echo $variaveis[3];

$informacao['nome'] = 'Guilherme';

$informacao['idade'] = 23;

$informacao['cidade'] = 'Florianópolis';

echo $informacao['nome'];

echo '<br />';

echo $informacao['idade'];

echo '<br />';

echo $informacao['cidade'];

?>

**Aula07 – Concatenação**

<?php

$frase = 'Meu nome é ';

$nome = 'Guilherme';

$idade = 23;

define('NOME', 'João');

//echo $frase.$nome.$idade;

//echo 'Olá, meu nome é '.$nome;

//echo "Meu nome é $nome";

echo "<div class=\"$nomedaclasse\">Meu conteúdo da div</div> ";

?>

**Aula08 – Operações matemáticas**

<?php

$numero = 10;

$numero2 = 10-4;

$resultado = $numero+$numero2;

echo $resultado;

?>

<?php

$numero = 10;

$numero2 = 2;

$resultado = $numero/$numero2;

echo $resultado;

?>

<?php

$numero = 10;

$numero2 = 2;

$resultado = $numero+$numero2;

$resultado2 = 10/5;

//echo $resultado;

//$expressao = 10 + (20 \* 2);

$expressao = 20 \* (2 + 10);

$resto = 4 % 2;

// SEW O RESTO FO IGUAL A 0

// SABEMOS QUE TEMOS UM NUMERO PAR

echo $resto;

?>

**­­­­­­­­­­­­­­­­­­­­­­­­­­Aula09 – Condições**

<?php

$variavel1 = 'Guilherme';

$variavel2 = 'João';

if($variavel1 == $variavel2){

echo '<div style="width:300px;height:300px;background:red;"></div>';

}else{

echo 'A condição não bate';

}

?>

<?php

$variavel1 = 'Guilherme';

$variavel2 = 'João';

if($variavel1 == $variavel2){

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title></title>

</head>

<body>

Este é meu Site!

</body>

</html>

<?php

}else{

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title></title>

</head>

<body>

Outro Site!

</body>

</html>

<?php

}

?>

<?php

$var = 10 + 5;

$var1 = '10 + 5';

if($var == $var1){

echo 'verdade';

}else if ($var != $var1){

echo 'Verdade na condição else if';

}else{

echo 'falso';

}

?>

**Aula10 – Operadores**

<?php

$valor = 10;

$valor2 = 9;

//Um sinal de igual quer dizer que estamos atribuindo um valor.

//Sempre retorna verdadeiro.

/\*

if ($nome = $nome2) {

echo $nome;

}

\*/

//Apenas confere se o valor é igual.

/\*if ($nome == $nome2){

echo 'verdade';

}\*/

//Conferimos para ver se é diferente.

/\*if($nome != $nome2){

echo 'é diferente';

}

if($nome === $nome2){

echo 'verdade';

}else{

echo 'não são identicos';

}\*/

//Comparamos para ve se são diferentes (tipo e valor)

if($valor !== $valor2){

echo 'verdade';

}

?>

**Aula11 – Operadores matemáticos e lógica**

<?php

/\*

$numero1 = 10;

$numero2 = 11;

if($numero1 < $numero2){

//menor que

}else{

}

if($numero1 <= $numero2){

//menor igual que

echo 'execute';

}

if($numero1 > $numero2){

//maior que

}else{

}

if ($numero1 >= $numero2) {

//maior igual que

echo 'execute';

}\*/

$var1 = 'valor1';

$var2 = 'valor2';

$var3 = 'valor1';

//Operador logico 'E'

/\*

if(($var1 == $var2) && ($var2 == $var3)){

echo 'verdade';

}\*/

//Operador logico 'OU'

if(($var1 == $var2 || $var1 == $var3) && $var1 == $var2){

echo "verdade";

}

?>

**Aula12 – Loopings**

<?php

/\*

//Laço de repetição 'for'

for($contador=10;$contador>=0;$contador--){

echo 'Olá, Mundo!';

echo '<br />';

echo $contador;

echo "<hr>";

}

//Laço de repetição 'while'

$contador = 0;

while ($contador <= 10) {

// code...

echo "Alô mundo";

echo '<hr>';

$contador+=1;

}\*/

//Laço de repetição 'do'

$contador = 0;

do{

echo 'Olá Mundo!';

echo '<br />';

$contador++;

}while ($contador < 10);

?>

**­­­­­­­­­­­­­­­­­­­­­­Aula13 – Comentários no código**

<?php

//variável que vou delcarar o nome da pessoa. (quando for apenas uma linha)

$nome = 'Guilerme';

/\*

Utilizar este tipo de

comentário quando for para mais de uma linha

é muito importante comentar para documentar

o código

\*/

$idade = 24;

#outra forma de fazer comentário

?>

**Módulo 6 - Aula01 –– Foreach e for para arrays – Nível Intermediário**

<?php

$arr = array('Guilherme','João','Felipe','Mario');

/\*

foreach($arr as $key => $value) {

echo $key;

echo '=>';

echo $value;

echo '<hr>';

}

\*/

$total = count($arr);

for($i = 0; $i < $total; $i++){

echo $arr[$i];

echo '<br />';

}

?>

**Aula02 – Array multidimensional**

<?php

//Array single

$arr = ['Guilherme', 'João'];

$arr = array('Guilherme','chave2'=>'João');

$arr[0] = 'Guilherme';

$arr[] = 'João';

//Arrays Multidimencionais

//$arr2 = array(array('Guilherme','João'),array(23,45));

//$arr2[0] = array('chave1'=>'Guilherme','João');

$arr2[0]['chave1'] = 'Felipe';

echo $arr2[0]['chave1'];

?>

**Aula03 – Die e sleep**

<?php

//Durma o código por x segundos.

/\*

sleep(2);

echo 'Olá, Mundo!';

sleep(2);

echo ' => ';

sleep(2);

echo 'Bye!'

\*/

$nome = 'Gui';

if($nome == 'João'){

//continuamos o código

echo 'tudo certo!';

}else{

die("O script parou de ser executado!");

}

sleep(2);

echo 'Olá, Mundo!';

sleep(2);

echo ' => ';

sleep(2);

echo 'Bye!';

?>

**Aula04 – Entendendo funções nativas e criando funções**

<?php

/\*

function mostrarNome($nome,$idade){

echo '<h2>Nome é: </h2>'.$nome;

echo '<hr>';

echo "<h2>Idade: </h2>$idade";

}

mostrarNome('João',23);\*/

function calculadora($numero1 = 10,$numero2 = 5,$numero3 = 0){

echo ($numero1+$numero2+$numero3);

}

//calculadora(20,20,30);

function verdade(){

return true;

}

function retornarString(){

return 'Guilherme';

}

echo retornarString();

$variavel = verdade();

?>

**Aula05 – Utilizando include e date**

**Date**

<?php

//Trabalhando com datas

date\_default\_timezone\_set('America/Sao\_Paulo');

$data = date('d/m/Y H:i:s', time()+(60\*10));

echo $data;

?>

**Include**

Index.php

<?php

$titulo\_site = 'Titulo do meu site';

include('header.php');

?>

<h1>Meu conteudo da Home!</h1>

<?php

include('footer.php');

?>

header.php

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title><?php echo $titulo\_site; ?></title>

</head>

<body>

Aqui é meu header

footer.php

Aqui é meu footer

</body>

</html>

**Aula06 – Funções para Strings**

<?php

$conteudo = 'Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industrys standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.';

//serve para recortar uma string.

//echo substr($conteudo,0,20);

$nome = 'Guilherme Cherem Grillo';

$nomes = explode(' ', $nome);

//print\_r($nomes);

//Serve para juntar um array com um delimitador.

//No caso o espaço.

$nomes = implode(' ', $nomes);

//echo $nomes;

//Strips\_tags serve para retirar todo o codigo HTML.

$conteudo = '<h1>Guilherme</h1>Outra coisa';

//echo strip\_tags("$conteudo");

//htmlentities que mostra o codigo html na pagina.

echo htmlentities('<div></div>');

?>

**Aula07 – Switch, continue e break**

<?php

//$nome = 'João';

/\*

switch ($nome) {

case 'Guilherme':

echo "Minha variável é Guilherme";

break;

case 'João':

echo 'Minha variável é João';

break;

}

\*/

//Break para o loop: for, while, do, foreach e switch;

//Break tambem utilizamos para switch;

//Continue para loop: for, while, do e foreach;

for($i = 0; $i < 10; $i++){

if ($i == 5 || $i === 9)

continue;

echo "$i";

echo '<hr>';

/\*

if($i == 5)

break;

\*/

}

/\*

$var = false;

if ($var == true)

$nome = 'Guilherme';

$idade = 23;

echo $idade;

echo '<hr>';

echo $nome;

\*/

?>

**Aula08 – Funções para Manipular arrays**

<?php

//Serve para unir 1 ou mais arrays;

/\*

$array1 = array("chave1"=>"valor1","chave2"=>"valor2");

$array2 = array("chave3"=>"valor3","chave4"=>"valor4");

$array3 = array("chave5"=>"valor5");

$result = array\_merge($array1, $array2, $array3);

print\_r($result);

\*/

//array intersect key serve para retornar valores com a mesma chave em 1 ou mais arrays.

/\*

$array1 = array("chave1"=>"valor1","chave2"=>"valor2");

$array2 = array("chave1"=>"valor3","chave2"=>"valor4");

print\_r(array\_intersect\_key($array1, $array2))

\*/

//array map serve para aplicar uma função em todos os valors do array.

/\*

$arr = ['<p>guilherme</p>','joao','<h1>fabricio</h1>'];

print\_r(array\_map('strip\_tags', $arr));

\*/

?>

**Aula09 – Formulários, GET e POST**

<?php

echo '<form>

<input type="text" />

</form';

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Formulário</title>

</head>

<body>

<?php

/\*

if (isset($\_GET['acao'])) {

$nome = @$\_GET['nome'];

$email = @$\_GET['email'];

echo $nome;

echo'<br />';

echo $email;

}

\*/

if (isset($\_POST['acao'])) {

$nome = @$\_POST['nome'];

$email = @$\_POST['email'];

echo $nome;

echo'<br />';

echo $email;

}

?>

<form method="post">

<input type="text" name="nome" />

<input type="text" name="email" />

<input type="submit" name="acao" value="Enviar" />

</form>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Formulário</title>

</head>

<body>

<?php

if (isset($\_POST['acao'])) {

echo $\_POST['nome'];

}

?>

<form method="post">

<select name="nome">

<option value="Guilherme">Guilherme</option>

<option value="João">João</option>

</select>

<input type="submit" name="acao" value="Enviar" />

</form>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Formulário</title>

</head>

<body>

<?php

/\*

if (isset($\_GET['acao'])) {

$nome = @$\_GET['nome'];

$email = @$\_GET['email'];

echo $nome;

echo'<br />';

echo $email;

}

if (isset($\_POST['acao'])) {

echo $\_POST['numero1'] + $\_POST['numero2'];

}

\*/

if (isset($\_POST['acao'])) {

foreach ($\_POST['valor'] as $key => $value) {

echo $key;

echo '=>';

echo $value;

echo '<hr>';

}

}

?>

<form method="post">

<select name="nome">

<option value="Guilherme">Guilherme</option>

<option value="João">João</option>

</select>

<input type="checkbox" name="valor[]" value="10">10

<input type="checkbox" name="valor[]" value="20">20

<input type="checkbox" name="valor[]" value="30">30

<input type="checkbox" name="valor[]" value="40">40

<input type="submit" name="acao" value="Enviar" />

</form>

</body>

</html>

**Aula10 – Sessões e cookies**

Index.php

<?php

session\_start();

$\_SESSION['nome'] = '';

?>

<?php

session\_start();

$\_SESSION['nome'] = 'Guilherme';

$\_SESSION['idade'] = '24';

echo $\_SESSION['nome'];

echo $\_SESSION['idade'];

?>

Cookies

<?php

session\_start();

setcookie('nome','Guilherme',time() + (60+60+24),'/');

echo $\_COOKIE['nome'];

?>

Index2.php

<?php

session\_start();

if(isset($\_SESSION['nome'])){

echo $\_SESSION['nome'];

}

?>

<?php

session\_start();

$\_SESSION['nome'] = 'Gui';

$\_SESSION[0] = 'Gui';

session\_destroy();

?>

**Aula11 – Syntaxe alternativa no PHP**

<?php

$nome = 'Guilherme';

if ($nome == 'Guilherme'):

echo "Verdade";

endif;

echo '<br />';

$contador = 0;

while($contador < 10):

$contador++;

echo "Oi";

echo '<br />';

endwhile;

$array = array();

foreach ($array as $key => $value):

endforeach;

for ($i=0; $i < 10; $i++):

//aqui vai o código!

endfor;

?>

**Módulo7 – Aula01 – Introdução a orientação a objetos**

<?php

class Pessoa{

//objeto pessoa

private $nome = 'Guilherme';

private $idade = '23';

private $peso = '70kg';

public function crescer(){

$this->comer();

echo 'estou crescendo';

}

private function comer(){

echo 'estou comendo';

}

}

//Instanciar

$pessoa = new Pessoa;

$pessoa2 = new Pessoa;

$pessoa->crescer();

?>

**Aula02 – Public, private e static**

Index.php

<?php

include('Exemplo.class.php');

//Instancia de objeto

$exemplo = new Exemplo();

$exemplo->setVar1('Guilherme');

echo $exemplo->pegaVar1();

//$exemplo->var2 = 'Guilherme';

echo '<hr />';

$exemplo2 = new Exemplo;

$exemplo2->setVar1('João');

echo $exemplo2->pegaVar1();

//$exemplo2->var2 = 'João';

/\*

echo $exemplo->var2;

echo '<br />';

echo $exemplo2->var2;

Exemplo::$var3 = 'Outra Variável';

echo Exemplo::$var3;

Exemplo::metodoEstatico();

\*/

?>

Exemplo.class.php

<?php

/\*\*

\* Esta Classe é de exemplo

\*/

//public funciona em qualquer lugar!

//private só conseguimos acessar dentro da classe!

class Exemplo

{

private $var1;

public $var2;

/\*

public static $var3 = 'statico';

public function metodo(){

}

public static function metodoEstatico(){

//echo 'metodo estatico';

}

private function metodo2(){

}

\*/

public function setVar1($var1){

$this->var1 = $var1;

}

public function pegaVar1(){

return $this->var1;

}

}

?>

**Aula03 – Final class, heranças, protected**

<?php

//final class Filha{} - Não pode ser Herdada

class Filha{

/\*protected function funcaoTeste(){

echo 'chamando função teste';

}\*/

private function funcaoTeste(){

echo 'chamando funcao teste';

}

public function mostrarOla(){

$this->funcaoTeste();

echo '<br />';

echo "Olá, Mundo!";

}

}

class Pai extends Filha{

public function mostrarOla(){

parent::mostrarOla();

echo '<br />';

echo 'nova função';

}

public function mostrarTchau(){

echo 'Tchau Mundo!';

echo '<br />';

$this->funcaoTeste();

}

}

/\*

$pai = new Pai;

$pai->mostrarTchau();

echo '<hr />';

\*/

$pai = new Pai;

$pai->mostrarOla();

//$pai->mostrarTchau();

//$filha->mostrarOla();

?>

**­­­­­­­­­­­­­­­­­­­­­­­Aula04 – Abstract, self e mais conceitos**

<?php

abstract class teste{

public $var;

public function func1(){

echo "chamando função 1";

}

abstract function func2();

}

class Principal2{

public static function outroMetodoEstatico(){

echo 'meu outro metodo';

}

}

class Principal extends teste{

public function func2(){

echo 'Estou delcarando ofialmente um metodo abstrato';

}

public static function metodoestatico(){

echo 'metodo estatico';

}

public function funcao(){

//Principal::metodoestatico();

//self::metodoestatico();

Principal2::outroMetodoEstatico();

}

}

$principal = new Principal;

$principal->funcao();

//Principal::metodoestatico();

/\*$principal->func1();

echo '<br />';

$principal->func2();\*/

?>

**Aula05 – Interface e implements**

Index.php

<?php

include('Interface1.php');

class Teste implements Interface1{

public function printOnScreen($par){

echo $par;

}

}

$teste = new Teste;

$teste->printOnScreen('Ola, Mundo!');

?>

Interface1.php

<?php

interface Interface1{

public function printOnScreen($par);

}

?>

**Aula06 – Métodos mágicos**

Index.php

<?php

include('Class1.php');

$teste = new Class1('Guilherme',23);

echo $teste->getNome();

echo $teste->getIdade();

?>

Class1.php

<?php

class Class1

{

private $nome;

private $idade;

function \_\_construct($nome,$idade)

{

$this->nome = $nome;

$this->idade = $idade;

}

public function getNome(){

return $this->nome;

}

public function getIdade(){

return $this->idade;

}

}

?>

**Aula07 – Namespace e use Keywords**

Index.php

<?php

include('Class1.php');

include('Class2.php');

use \Sessao1\Class1 as classe\_exemplo;

$class1 = new classe\_exemplo();

//$class2 = new Class2();

?>

Class1.php

<?php

namespace Sessao1;

use Sessao2\Class2;

class Class1

{

function \_\_construct()

{

new \Sessao2\Class2();

echo 'Class1 instanciada';

}

}

?>

Class2.php

<?php

namespace Sessao2;

class Class2

{

function \_\_construct()

{

echo 'Class2 instanciada';

}

}

?>

**Aula08 – Autoload**

Index.php

<?php

include('config.php');

spl\_autoload\_register('myAutoLoad');

//new Utilidades();

new Home\Inicial();

?>

Config.php

<?php

function myAutoLoad($class){

$class = str\_replace('\\', '/',$class);

if(file\_exists('classes/'.$class.'.php')){

include('classes/'.$class.'.php');

}

}

?>

Utilidades.php

<?php

class Utilidades

{

function \_\_construct()

{

echo 'Classe Utilidades foi chamada com sucesso!';

}

}

?>

Inicial.php

<?php

namespace Home;

class Inicial

{

function \_\_construct()

{

echo 'Classe Inicial chamado com sucesso!';

}

}

?>

**Aula09 – Constantes no PHPOO**

<?php

class minhaClasse

{

const VALOR = 300;

public function \_\_construct(){

echo self::VALOR;

}

}

echo minhaClasse::VALOR;

?>

**Módulo8 – Aula01 – Usando PDO e criando banco**

<?php

$pdo = new PDO('mysql:host=localhost;dbname=modulo\_8','root','');

?>

**Aula02 – Insert PDO**

<?php

date\_default\_timezone\_set('America/Sao\_Paulo');

$pdo = new PDO('mysql:host=localhost;dbname=modulo\_8','root','');

if(isset($\_POST['acao'])){

$nome = $\_POST['nome'];

$sobrenome = $\_POST['sobrenome'];

$momento\_registro = date('Y-m-d H:i:s');

$sql = $pdo->prepare("INSERT INTO `clientes` VALUES (null,?,?,?)");

$sql->execute(array($nome,$sobrenome,$momento\_registro));

echo "Cliente inserido com sucesso!";

}

?>

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Cadastro no Banco</title>

</head>

<body>

<form method="post">

<input type="text" name="nome" required />

<input type="text" name="sobrenome" required />

<input type="submit" name="acao" value="Enviar!" />

</form>

</body>

</html>

**Aula03 – UPDATE e Condições nas queries**

<?php

$pdo = new PDO('mysql:host=localhost;dbname=modulo\_8','root','');

$id = 2;

//OR = `ou` - AND = `e`

$sql = $pdo->prepare("UPDATE `clientes` SET nome='Cezar',sobrenome='Santana de Almeida' WHERE id=$id");

if($sql->execute()){

echo "Meu cliente foi atualizado com sucesso!";

}

?>

**Aula04 – Delete um PDO**

<?php

$pdo = new PDO('mysql:host=localhost;dbname=modulo\_8','root','');

$id = 2;

//OR funciona como `ou` | AND funciona como `e`

$sql = $pdo->prepare("DELETE FROM `clientes` WHERE id=?");

if($sql->execute(array($id))){

echo "Meu cliente foi Deletado com sucesso!";

}

?>

**Aula05 – Inner Join e mais conceitos**

<?php

$pdo = new PDO('mysql:host=localhost;dbname=modulo\_8','root','');

/\*

$sql = $pdo->prepare("SELECT \* FROM posts WHERE categoria\_id = ?");

$sql->execute(array($\_GET['categoria\_id']));

$info = $sql->fetchAll();

for($i = 0; $i < count($info); $i++){

echo 'Título: '.$info[$i]['titulo'];

echo '<br />';

echo 'Noticia: '.$info[$i]['conteudo'];

echo "<hr>";

}

\*/

/\*

$sql = $pdo->prepare("SELECT \* FROM categorias");

$sql->execute();

$info = $sql->fetchAll();

foreach ($info as $key => $value) {

$categoria\_id = $value['id'];

echo "Exibindo posts de: ".$value['nome'];

echo '<br />';

$sql = $pdo->prepare("SELECT \* FROM posts WHERE categoria\_id = $categoria\_id");

$sql->execute();

$infoPosts = $sql->fetchAll();

foreach ($infoPosts as $key => $value) {

echo 'Título: '.$value['titulo'];

echo '<br />';

echo 'Noticia: '.$value['conteudo'];

echo "<hr>";

}

}

\*/

/\*

echo "<pre>";

print\_r($info);

echo '</pre>';

\*/

$sql = $pdo->prepare("SELECT `posts`.\*,`categorias`.\*,`posts`.`id` AS `post\_id` FROM `posts` INNER JOIN `categorias` ON `posts`.`categoria\_id` = `categorias`.`id`");

$sql->execute();

$info = $sql->fetchAll(PDO::FETCH\_ASSOC);

echo "<pre>";

print\_r($info);

echo "</pre>";

?>

**Aula06 – Group, Order e Limit**

<?php

$pdo = new PDO('mysql:host=localhost;dbname=modulo\_8','root','');

$sql = $pdo->prepare("SELECT \* FROM `clientes` GROUP BY `cargo`");

//$sql = $pdo->prepare("SELECT \* FROM `clientes` ORDER BY nome ASC LIMIT 0,3");

$sql->execute();

$clientes = $sql->fetchAll();

foreach ($clientes as $key => $value) {

echo $value['nome'];

echo '<hr>';

}

?>

**Aula07 – LEFT e RIGHT Join**

<?php

$pdo = new PDO('mysql:host=localhost;dbname=modulo\_8','root','');

$sql = $pdo->prepare("SELECT \* FROM `clientes` LEFT JOIN `cargos` ON `clientes`.`cargo` = `cargos`.`id`");

$sql->execute();

$clientes = $sql->fetchAll();

foreach ($clientes as $key => $value) {

echo $value['nome'];

echo '|';

echo $value['nome\_cargo'];

echo '<hr>';

}

?>

Aula08 – Segurança, import/export e instancia avançada do PDO

<?php

define('HOST','localhost');

define('DB', 'modulo\_8');

define('USER', 'root');

define('PASS', '');

//tratamento de erro para não deixar mostrar dados da senha do servidor SQL.

try{

$pdo = new PDO('mysql:host='.HOST.';dbname='.DB,USER,PASS,array(PDO::MYSQL\_ATTR\_INIT\_COMMAND => "SET NAMES utf8"));

$pdo->setAttribute(PDO::ATTR\_ERRMODE,PDO::ERRMODE\_EXCEPTION);

}catch(Exception $e){

echo 'erro ao conectar';

}

/\*

$sql = $pdo->prepare("SELECT \* FROM `posts9`");

$sql->execute();

\*/

?>

**Aula09 – Lock tables**

Index.php

<?php

$pdo = new PDO('mysql:hos=localhost;dbname=modulo\_8','root','');

$pdo->exec("LOCK TABLES `clientes` WRITE");

sleep(10);

?>

Index2.php

<?php

$pdo = new PDO('mysql:hos=localhost;dbname=modulo\_8','root','');

$sql = $pdo->prepare("SELECT \* FROM `clientes`");

$sql->execute();

$usuarios = $sql->fetchAll();

foreach ($usuarios as $key => $value) {

echo $value['nome'];

echo '<hr>';

}

?>

**Aula10 – Pesquisa avançada com LIKE**

<?php

$pdo = new PDO('mysql:host=localhost;dbname=modulo\_8','root','');

$sql = $pdo->prepare("SELECT \* FROM `clientes` WHERE email LIKE '%g%'");

$sql->execute();

$emails = $sql->fetchAll();

print\_r($emails);

?>

**Aula11 – Entendendo IN e BETWEEN**

<?php

$pdo = new PDO('mysql:host=localhost;dbname=modulo\_8','root','');

$sql = $pdo->prepare("SELECT \* FROM clientes WHERE id IN (1,2,3)");

$sql->execute();

$emails = $sql->fetchAll();

print\_r($emails);

?>

<?php

$pdo = new PDO('mysql:host=localhost;dbname=modulo\_8','root','');

$sql = $pdo->prepare("SELECT \* FROM clientes WHERE data BETWEEN '2024-01-01' AND '2024-05-06'");

$sql->execute();

$emails = $sql->fetchAll();

print\_r($emails);

?>

**Aula12 – Manipulando arquivos**

<?php

//vamos criar um arquivo novo.

//$content = "Meu novo conteúdo criado através do PHP.\r\nOutra linha.\r\nMais uma Linha.";

//file\_put\_contents('file.txt', $content);

$content = file\_get\_contents('file.txt');

echo nl2br($content);

?>

//criar arquivos acrescentando conteúdo no texto interno

<?php

for($i = 0; $i < 100; $i++){

$conteudo = 'Meu contúdo do arquivo'.$i;

file\_put\_contents('file'.$i.'.txt',$conteudo);

}

?>

//remover os arquivos criados

<?php

for($i = 0; $i < 100; $i++){

//$conteudo = 'Meu contúdo do arquivo'.$i;

unlink('file'.$i.'.txt');

}

?>

**Aula13 – Manipulando pastas**

<?php

//criar pasta

/\*

mkdir('pasta');

if (is\_dir('pasta')) {

echo 'É uma pasta válida';

}else{

echo "Pasta não existe";

}\*/

//deletar a pasta

//rmdir('pasta');

//verificar conteudo das pastas

if ($handle = opendir('Pasta')) {

/\* Esta é a forma correta de percorrer o diretório \*/

while ($file = readdir($handle)) {

if($file == '.'|| $file == '..'){

continue;

}

/\*

if(is\_dir('pasta/'.$file) == false){

//é um arquivo

}else{

//uma pasta

}\*/

echo "$file\n";

echo '<br />';

}

closedir($handle);

}

?>

**Módulo\_9 – Aula01 – Lendo e escrevendo com XML**

Index.php

<?php

$xml = simplexml\_load\_file('arquivo.xml');

//Ler arquivo XML

//echo $xml->informacoes2->item->nome;

//echo $xml->informacoes->item->idade;

//print\_r($xml);

//Criando XML a partir de um array

//$arr = ['Guilherme' =>'nome','24'=>'idade'];

excreverXML(array('Guilherme'=>'nome','24'=>'idade','futebol,tennis'=>'esportes'),'arquivo.xml');

/\*$xml = new SimpleXMLElement('<root/>');

array\_walk\_recursive($arr, array ($xml, 'addChild'));

file\_put\_contents('arquivo.xml',$xml->asXML());\*/

//Lendo o XML

$content = simplexml\_load\_file('arquivo.xml');

echo $content->esportes;

function excreverXML($arr,$nomedoarquivo){

$xml = new SimpleXMLElement('<root/>');

array\_walk\_recursive($arr, array ($xml, 'addChild'));

file\_put\_contents('arquivo.xml',$xml->asXML());

}

?>

Arquivo.xml

<?xml version="1.0"?>

<root>

<nome>Guilherme</nome>

<idade>24</idade>

<esportes>futebol,tennis</esportes>

</root>

**Aula02 – Lendo e escrevendo com JSON**

Index.php

<?php

/\*

$json = '{"a":"Guilherme","b":{"0":"Felipe","1":"João"},"c":3,"d":4,"e":5}';

//Decodificamos o json

$obj = json\_decode($json);

$arr = json\_decode($json, true);\*/

//echo $arr['b']['1'];

//echo $obj->a;

//var\_dump(json\_decode($jason, true));

/\*

$arr = ['nome'=>'Guilherme','cidade'=>'Floripa'];

$json = json\_encode($arr);

echo $json;\*/

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title></title>

</head>

<body>

<script type="text/javascript" src="https://code.jquery.com/jquery-3.7.1.min.js"></script>

<script type="text/javascript">

$(function(){

$.ajax({

url:'request.php',

dataType:'json'

}).done(function(data){

console.log(data.nome);

})

})

</script>

</body>

</html>

Request.php

<?php

$arr = ['nome'=>'Guilherme','cidade'=>'Floripa'];

$json = json\_encode($arr);

echo $json;

?>

**Modulo\_10 – Aula01 – Google API**

<form method="post">

<input type="text" name="adress">

<input type="submit" name="acao" value="Enviar">

</form>

<?php

if(isset($\_POST['acao'])){

$url = urlencode($\_POST['adress']);

$str = file\_get\_contents('https://maps.googleapis.com/maps/api/geocode/json?address='.$url.'&sensor=true&key=AIzaSyDfn09TXYb9PY8fhjz67j2wTmECpkI-uRY');

$endereco = json\_decode($str);

/\*

echo '<pre>';

print\_r($endereco);

echo '</pre>';

\*/

echo 'Nome do Bairro: ';

echo $endereco->results[0]->address\_components[1]->short\_name;

echo '<hr>';

echo 'Nome correto da rua: ';

echo $endereco->results[0]->address\_components[0]->short\_name;

}

?>

**Aula02 – CURL e criando web service**

Index.php

<?php

//Aula CURL

$endereco = urlencode('coqueiros');

$curl = curl\_init();

curl\_setopt($curl, CURLOPT\_URL,"https://maps.googleapis.com/maps/api/geocode/json?address=".$endereco."&sensor=true&key=AIzaSyDfn09TXYb9PY8fhjz67j2wTmECpkI-uRY");

//curl\_setopt($curl, CURLOPT\_URL,"http://localhost/modulo\_010/aula\_02/request.php");

curl\_setopt($curl,CURLOPT\_POST,1);

//na vida real voce pode usar passando como um array e usar a função http\_build\_query...

curl\_setopt($curl, CURLOPT\_POSTFIELDS,http\_build\_query(array('request'=>'name\_return')));

//receive server response...

curl\_setopt($curl, CURLOPT\_RETURNTRANSFER,true);

$server\_output = curl\_exec($curl);

curl\_close($curl);

echo $server\_output;

//requisição com CURL no server local

/\*

$resultado = json\_decode($server\_output);

echo $resultado->titulo[0];

echo '<br />';

echo $resultado->conteudo[0];

\*/

?>

Request.php

<?php

if(isset($\_POST['request'])&& $\_POST['request'] == 'name\_return'){

//seleciona as noticias...

$noticias = array('titulo'=>array('noticia 1','noticia 2'),'conteudo'=>array('conteudo da noticia 1','conteudo da noticia 2'));

die(json\_encode($noticias));

}

?>

**Aula03 – Login Facebook PHP**

Index.php

<?php

include('config.php');

if(isset($accessToken)){

if(isset($\_SESSION['facebook\_access\_token'])){

$fb->setDefaultAccessToken($\_SESSION['facebook\_access\_token']);

}else{

$\_SESSION['facebook\_access\_token'] = (string)$accessToken;

$oAuth2Client = $fb->getOAuth2Client();

$longLivedAccessToken = $oAuth2Client->getLongLivedAccessToken($\_SESSION['facebook\_access\_token']);

$\_SESSION['facebook\_access\_token'] = (string)$longLivedAccessToken;

$fb->setDefaultAccessToken($\_SESSION['facebook\_access\_token']);

}

if(isset($\_GET['code'])){

header('Location: ./');

}

try{

$profileRequest = $fb->get('/me?fields=name,first\_name,last\_name,email,link,gender,locale,picture');

$fbUserProfile = $profileRequest->getGraphNode()->asArray();

}catch(FacebookResponseException $e){}

//print\_r($fbUserProfile);

$fbUserData = [

'oauth\_provider' => 'facebook',

'oauth\_uid'=>$fbUserProfile['id'],

'first\_name'=>$fbUserProfile['first\_name'],

'last\_name'=>$fbUserProfile['last\_name'],

'email'=>$fbUserProfile['email'],

'imagem'=>$fbUserProfile['picture']['url']

];

$userData = $fbUserData;

$\_SESSION['userData'] = $fbUserData;

$logoutUrl = $helper->getLogoutUrl($accessToken, $redirectUrl.'logout.php');

if(!empty($userData)){

$output = '';

$output.="<h1>Nome: $userData[first\_name]</h1>";

$output.="<h1>Sobrenome: $userData[last\_name]</h1>";

$output.="<h1>E-mail: $userData[email]</h1>";

$output.='<br /><img src="'.$userData['imagem'].'" />';

$output.='<br /><a href="'.$logoutUrl.'">loggout</a>';

}else{

$output = '<h1 style="color:red">Ocorreu um erro!</h1>';

}

}else{

$loginUrl = $helper->getLoginUrl($redirectUrl,$fbPermission);

$output = '<a href="'.$loginUrl.'">Fazer login com facebook PHP</a>';

}

echo $output;

?>

Config.php

<?php

session\_start();

include('facebook-php-sdk/autoload.php');

use Facebook\Facebook;

use Facebook\Exceptions\FacebookResponseException;

use Facebook\Exceptions\FacebookSDKException;

$appId = '1004288041286907';

$appSecret = '4ba4fdac306b5229cdf6ffe94dc8a6cc';

$redirectUrl = 'http://localhost/login-facebook/';

$fbPermission = array('email');

$fb = new Facebook(array(

'app\_id'=> $appId,

'app\_secret'=> $appSecret,

'default\_graph\_version' => 'v3.0'

));

$helper = $fb->getRedirectLoginHelper();

try{

if(isset($\_SESSION['facebook\_access\_token'])){

$accessToken = $\_SESSION['facebook\_access\_token'];

}else{

$accessToken = $helper->getAccessToken();

}

}catch(FacebookResponseException $e){}

?>

Logout.php

<?php

include('config.php');

unset($\_SESSION['facebook\_access\_token']);

unset($\_SESSION['userData']);

session\_destroy();

header('Location: index.php');

?>

**Aula04 – Email marketing API**

Código muito grande – Apenas criado arquivo index.php

**Aula05 – Instagram API**

<?php

$request = file\_get\_contents('https://www.instagram.com/cezinhoo/?\_\_a=1');

$object = json\_decode($request);

$imagens = $object->user->media->nodes;

for($i = 0; $i < 10; $i++){

echo '<img src= "'.$imagens[$i]->thumbnail\_src.'" />';

}

?>

**Aula06 – Login Instagram**

Arquivo muito grande

**Módulo 11 – Aula01 /02 – Instalando, entendendo e instalando dependências no Composer**

**Módulo 12 – Aula01 – Importando BootStrap**

<!doctype html>

<html lang="pt-br">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Projeto Dev Web</title>

<!-- bootstrap -->

<link href="css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<nav class="navbar navbar-expand-lg bg-body-tertiary">

<div class="container-fluid">

<a class="navbar-brand" href="#">Danki Code</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="#">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="contato">Contato</a>

</li>

<li class="nav-item">

<a class="nav-link" href="sobre">Sobre</a>

</li>

<li class="nav-item dropdown">

<ul class="dropdown-menu">

<li><a class="dropdown-item" href="#">Action</a></li>

<li><a class="dropdown-item" href="#">Another action</a></li>

<li><hr class="dropdown-divider"></li>

<li><a class="dropdown-item" href="#">Something else here</a></li>

</ul>

</li>

</ul>

<form class="d-flex" role="search">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

</body>

</html>